```
*************************************************************
                                  ***
        /////////      ////   ////  ***  /////////
       IIIIIIIII/     IIII/  IIII/    ***IIIIIIII//
      IIIII/          IIII/  IIII/     IIII/  IIII/
      IIIII/          IIII/  IIII/     IIII/* IIII/
      IIIII/          IIII/  IIII/     IIII/***III/
      IIIII//////  // IIII///IIII/  // IIII///***I/  //
       IIIIIIIII/ II/  IIIIIIIII/  II/  IIIIIIII*** II/
                                                  ***
*************************************************************
                  CIRCULO DE USUARIOS DE QL
*************************************************************
```

Fanzine mensual independiente para usuarios de Sinclair QL y compatibles

---

AÑO 3 NUMERO 18 MARZO DE 1990

Estamos en el año 10 Después de Sinclair. Toda Hispania está ocupada por las legiones de PCs. ¿Toda? ¡No! Un puñado de irreductibles QLs resiste todavía y siempre ante el primitivo invasor...

Compilación de colaboraciones y distribución: Salvador Merino

Para recibir información sobre cómo recibir y/o colaborar en el fanzine, enviad un sobre franquedo y con vuestra dirección a: Marcos Cruz, Acacias 44, 28023 MADRID.

CONTENIDO

Portada de este número: CELESTINA CROSS

Con este número de CUQ se incluyen los programas siguientes:

- Celestina Cross en la isla de los monstruos.- Autor: Miguel Angel Monje Garcia.
- Dolor_bas.- Autor Miguel Frasquet.
- Dumps impresoras 8 y 24 agujas.- Cargar con Lrespr del Toolkit II. Autor: Salvador Merino.
- Antorchas.- (TRABAJO PARA TRANSLADAR A PC).

Material preparado para los próximos números:

- PROGRAMA TETRIS
- MAQUINA DE TRADUCCION
- Z88 FORTH SI EL QUANTUM Z80 SE DEJA DOMINAR

EDITORIAL

Con este número ya hace un año y medio que decidimos independizarnos y continuar la historia con un nuevo nombre, CUQ (sin ir más lejos).
Aunque este número en bytes es el más gordo hasta la fecha, no os hagais ilusiones, pues se trata de un extra por equivocación.
Hasta la fecha he tenido varias dificultades para ensamblar mi Z88 FORTH, y aún no he conseguido hacerlo correr como fue pensado en un principio. El problema reside en que el ensamblador Quantum Z80 en su versión diez y pico todavía no está lo suficientemente depurado, pues en la última versión he encontrado dos errores, pero tampoco se descarta cientos de errores en el listado assembler de mi Z88 FORTH.
El programa Celestina Cross pertenece a un ex-usuario de QL (ahora AMIGA, según me cuentan) de un club de Vizcaya que fotocopiaba los boletines de QLAVE (al irse a pique ésta, también se llevó al otro Club). El programa lo ha cedido un miembro del Club de Vizcaya, y desconozco con exactitud si habia autorización por parte del autor. De todas formas, el juego posee muy buenos gráficos, pero poco interés comercial en un mercado como el QL.
S. Merino

NOTICIAS

PROGRAMA RADIOFONICO

He descubierto por pura casualidad un programa de radio que trata de temas informáticos, el programa es los domingos por la mañana en radio 3. Es un programa con temas variados y hay una sección dedicada a la informática que suele empezar a las 12, el programa se llama "el Dominical".
En cuanto a la calidad que cada uno juzgue por si mismo, una de las veces que lo escuché había un grupo de gente que estaban metidos en el mundo de las BBS, la semana siguiente a la que estoy escribiendo esto tratarán sobre los productos que están apareciendo en el mercado español en discos CDrom.
Hasta la próxima.
Celestino Alvarez
21-1-90

CARTAS ABIERTAS

HARDWARE-BRICOLAGE

Hola amigos:

Esta colaboración es para aportar mi granito de arena al CUQ, pues tengo varias experiencias e ideas que quizás puedan servir a alguien.
Mi equipo está formado por un QL versión JM, algo antiguo e inglés original, pero actualizado con una Trump Card y una unidad de disco Micro Peripheral. La idea, en este caso para manitas del hardware-bricolage, es que todo el material disperso de alimentadores, cables y conexiones los he reunido en una sola caja metálica que permite su fácil manipulación e incluso transporte, como era mi caso al tener que cambiar el QL de sitio todas las semanas. La caja es una de las normales que venden en los establecimientos de electrónica por 2000-3000 ptas. en función del tamaño. Es preferible que la caja sea completamente de

plástico -aunque yo no la he encontrado lo suficientemente grande- puesto que una vez dentro todos los materiales, se crea una capa estática en la superficie de la caja que puede dejar colgado al QL. Yo he solucionado el problema añadiendole un plástico adhesivo que la hace más vistosa y "profesional". En la parte posterior se dejan cuatro aberturas: una lineal para el cable plano del controlador del disco, otra para la conexión al QL, otra para la conexión genral a red, y otra para el interruptor general. En la parte anterior se practica una abertura rectangular del tamaño exacto de la carátura del disco, de forma que por delante la caja presenta el aspecto general de las unidades centrales de cualquier otro ordenador.

El cableado interior supone uno general de potencia, con un enchufe que será el que se conecte a la red, uno de cuyos polos se conecta al interruptor posterior y el otro polo a un enchufe múltiple normal (si no se desea cortar los cables que sería lo más "profesional"). El segundo polo del interruptor va también a ese enchufe múltiple. Y en éste se conectan todos los transformadores que se tengan (el del QL y el de la unidad de disco), de esta forma el interruptor general apaga o enciende todo el equipo. Fácil verdad.

Se saca el cable de conexión al QL por atrás, y se conectan los demás cables de la unidad de disco. Se ha de sujetar todo el conjunto muy bien para evitar falsos contactos, bien con tornillos o tuercas de sujección para los cables sueltos. En mi caso, recorté las placas a medida sobre "corcho blanco" para que quedase cada cosa en su sitio. Si alguien es más "quisquilloso" respecto a la técnica puede elaborar una placa base en plástico sobre la que situar los transformadores sin carcasa, así como un interruptor con luz para controlar el on/off del aparato, e incluso un pequeño ventilador si se desea..

El conjunto así formado no emite más calor que sus partes aisladas, y yo lo llevo utilizando más de dos años sin problemas. Permite un fácil transporte e instalación, puesto que en un solo bloque está todo el material que puede situarse bajo el monitor o a un lado del QL. Al conectarlo no tiene más que tres cables (voltaje del QL, disco y general), y pulsando el interruptor posterior se puede desenchufar todo el equipo sin más.

Por otra parte, es un montaje bastante sencillo y barato que permite mucha mayor comodidad con el QL y sin llegar a la sofisticación de crearse toda una "especie de PC" con cajas, teclados alternativos, módulos de conexión, etc.

Respecto a las cintas de impresora tengo otra idea que puede ayudar a alguien, sobre todo por lo económico del tema. Yo tengo una Seikosha SP-1000AS, pero en general todas tienen cartuchos de cinta diferentes y además con precios nada baratos. La solución por la que yo he optado es cambiar yo mismo la cinta de cada cartucho por una cinta de máquina de escribir -si es de seda mucho mejor pues es la que más dura (500 folios) y deja la señal más limpia-. El procedimiento es sencillo: desarmar la carcasa de plástico de la cinta que viene a presión y sin tornilllos, sacar la cinta antigua, cortar a la cinta de máquina el canutillo de comienzo e introducirla por los rieles del cartucho para situarla en la posición "arrugada" que tenía la original, rebobinar progresivamente la cinta hasta acabarla y cortar el canutillo del otro extremo. Los cortes del principio y final de la cinta nueva se realizan en diagonal -como si fuese una cinta de video o de audio- y se pegan con algún tipo de pegamento de contacto sin superponer los dos extremos. Una hora después la cinta está lista para utilizar en la impresora, con más contraste y mayor duración que si se hubiese comprado en nuestro "computer-shop" favorito.

Otra idea, para quien maneje el Quill con documentos muy largos. Puesto que el programa no permite numerar hojas más allá de la 255, el truco consiste en modificar la cabecera o pié del documento para que escriba por ejemplo "2" y comenzar a numerar el documento con "56"; de esta forma al imprimir en la hoja saldrá la numeración completa "256".

Luis Valero, Málaga, 24/1/1990

AGRADECIMIENTO

Hola Salvador:

Quiero agradecerte las noticias que dabas en el ultimo boletin sobre los proyectos del tio Tebby al menos es un consuelo ver que no estamos abocados al exterminio absoluto y que en todo caso podremos continuar aunque sea como parientes pobres con nuestros QL.

He recibido un boletin de noticias del Quele Club en donde no hay nada nuevo y menos para ti. Me llama la atencion una "special edition" del Lighting en rom (8500) para el conector de rom's externo.

Lei en una de tus ultimas cartas en el CUQ que tenias un MSDOS 3.3 para el

QL y que tenias algun problema. Te envio el MSDOS 4.00 que me enviaron al comprar el emulador a Quele Club(19000). Es una version IBM y esta en castellano. Los discos te los quedas y ya me los envias con algo interesante.

Tambien me enviaron una fotocopia del QLWORL de Nov.89(cuanto tiempo sin ver una). El esta fotocopia se explica la forma o formas de hacer el QL a prueba de bomba (segun el titular). Como creo que es de interes general voy a intentar traducira y extractar lo mas interesante:

Comienza hablando de los motivos de los cuelgues del aparato, nada nuevo. Despues pasa a enumerar los remedios y por ultimo las opciones a estos remedios.

REMEDIOS:

-Cambiar el famoso 7805(es el que esta debajo del radiador que se encuentra a la derecha y atras de la PCB) por el no menos famoso 78S05. Este apaño es muy sencillo puesto que ambos son iguales, solo cambia el amperaje que ahora sera de 2 amp.

-Asegurarse de que el voltaje en la 68008 es 4.9v.(ignoro que hay que hacer en caso de que no lo sea).

-Poner un condensador de .1 mfd entre el centro y la pata derecha del 78S05.

-El cuarto remedio no lo se explicar debido a mi pesimo ingles, agradecere que los que dominen el idioma nos iluminen.

-Conectar un condensador de .01 mfd entre el pin 6 y el pin 5 del 8301, entondes otro de .001 mfd desde el pin 6 y pin 15 del 8301 y pin 35 del 68008.(Mejor que alguien lo explique con mas detalle).

-Soldar un condensador de .01 mfd en paralelo con un condensador de 'tantalum'(literal del ingles) de 10 mfd 10V y conectar el pin 20 y pin 40 con + al pin 40 del 8309 (coprocesador).

OPCIONES:

-Cambiar IC19, IC20, IC21 y IC27 por sus HCT equivalentes (sic.)

etc.(ya que mi ingles no llega a mas)

Creo que es interesante que alguien mas preparado se lea el articulo y lo explique un poco. Yo tengo cambiado el 7805 por el 78S05 y no tengo problemas de cuelgue pero la verdad es que en mas de una ocasion he notado la opresion en el cuello de solo pensar en esa posibilidad.

Saludos

Miguel Frasquet, Valencia,29/1/90.

OFERTA DE TRABAJO

Sorpresivamente, recibí en diciembre un llamada en la que se me preguntaba si podía pasar unos programas de QL a PC. Se trata de una empresa que había pedido consejo al respecto a los distribuidores del Z88, y éstos les habían dado mi teléfono, que estaba en su base de datos desde que les escribí hace meses solicitando información -que jamás recibí- sobre el mencionado Z88. El caso es que esta empresa tiene un QL desde hace bastante tiempo, que apenas usan, y un programa en SuperBasic para calcular antorchas (unas torres de las centrales térmicas) que querían pasar a PC. Me ofrecí a pasarles el texto del programa a formato MS-DOS en un disco de 5,25 para que pudieran cargarlo en algún tratamiento de textos y listarlo, pues el QL que tienen está averiado y no imprime, pero les dije, después de examinarlo, que el hacer la adaptación a algún BASIC de PC es cosa muy distinta. El programa está increiblemente enrevesado y sin documentar, lo cual hace una tarea casi imposible el ponerse a descifrarlo. Quedé en intentar contactar con alguien que pueda estar interesado en hacer el trabajo, mientras ellos intentaban examinar el listado y hacer un diseño oredenado de lo que debe hacer el programa.

En este número de CUQ se incluyen los listados del programa "antorchas" para que le eche un vistazo quien quiera. Uno de ellos estaba corrompido en el microdrive y tiene algunos REMs con la nota "zona corrompida" allí donde no ha sido posible salvar las líneas originales.

Si a alguno os interesa el trabajo de hacer la adaptación a PC, o informaros más en detalle sobre el tema, poneos en contacto con: Ricardo Carrilero, Prematécnica S.A., Orense 4-13, 28020 MADRID, teléfono 91-5557158.

Marcos C.M., Madrid, 25/1/1990

COMENTARIOS DE PROGRAMAS

PROGRAMA : PC CONQUEROR
DISTRIBUIDOR : DIGITAL PRECISION LTD.
222 The Avenue, Chingford, London E4 9SE (UK).
PRECIO : 89.95 LIBRAS (+5% GASTOS).

Hace ya un año que recibí mi copia de SOLUTION. Ahora he recibido mi upgrade PC CONQUEROR, lo cual podria tratarse de una locura por mi parte.

Como todos sabemos, PC CONQUEROR es un nuevo emulador de IBM PC basado solamente en software, y debido a ello su velocidad es su punto debil. No obstante, se ha conseguido aumentar la velocidad empleando nuevas técnicas revolucionarias, las cuales me atreveria a definir comparandolas con un coche convencional. Ya sabeis a que me refiero, freno, embrague, acelerador, cambio de marchas y volante.

¿Porqué esa definición? Muy fácil. PC CONQUEROR es un emulador IBM PC que hace trampas usando las facilidades que nos proporciona nuestro querido y brillante QDOS. A diferencia de SOLUTION, podemos cambiar en modo SUPERVISOR (en mi configuración pulsando la tecla Ñ, pues CTRL, SHIFT, CAPS LOCK y ALT, me da problemas) la prioridad del modo gráfico, modo texto y después de una llamada al BIOS.

Según el manual, cambiando las prioridades de los jobs se pueden conseguir velocidades muy interesantes. P.e.: un 85% de la velocidad de un XT con la combinación 4, 4, 96 (en mi QL con MS-DOS 4.0, solamente 80%) usando Advanced PC Tools v4.11 benchmark. La configuración por defecto (10,6,32) me da un 55%. Usando EXEC_W a cambio de EXEC se consigue un aumento del 2%. Aunque lo más interesante es que usando MS-DOS v3.2/v3.3 se consigue un 30% más de velocidad, pues son más rápidas que las recientes versiones de MS-DOS (Digital Research's DR-DOS es aún mucho más rápido).

En un Atari ST/QL emulator el aumento de velocidad es sobre 2.7x, y con el THOR XVI se consigue una velocidad similar. Y cuando llegue un 20 MHz+ 68020/68030 Super-QL (no un sueño imposible, según asegura Digital Precision), CONQUEROR será varias veces más rápido que la mayoria de los PCs.

Sobre lo de un Super-QL basado en un 68020, después de haber leido un comentario del Club Francés sobre el THOR XVI que me ha fotocopiado nuestro amigo y compañero de circulo Dasio (de Santiago Compostela), creo que podria tratarse de una realidad en muy corto espacio de tiempo. El THOR XVI es un ordenador basado en un diseño modular (igual que el viejo Apple II), lo que significa que puede actualizarse a 68020 a 16 MHz con solamente cambiar la tarjeta que contiene el 68000 por otra nueva, pudiendo seguir usando la tarjeta controladora de periféricos. Otra característica a destacar es que ahora el THOR XVI puede correr la QRAM y sus derivados (todo el software QJUMP), y existe la posibilidad de la instalación de un circuito anexo y cambio de ROMs para 1024x512 puntos en monocromo. Otra característica con el HD 6445 es desplazar la memoria para obtener resoluciones diferentes (p.e.: una paleta gráfica de 16 colores a elegir de 4096).

No sé si os acordais de las pruebas de velocidad en GW-BASIC de José Carlos de Prada (ver CUQ-8).

En el TEST1, SOLUTION necesitaba 218 segundos. CONQUEROR necesita solamente 178 segundos con la configuración defecto o la que da un 80% de velocidad XT.

En el TEST2, SOLUTION necesitaba 1938 segundos. En CONQUEROR hay problemas, necesita 2554 segundos en configuración defecto, 3488 segundos en configuración 80% XT. Me temo que para conseguir solamente la misma velocidad que en SOLUTION debemos encontrar la configuración ideal a base de probar valores (probando con otras configuraciones he conseguido rebajar algunos minutos).

Otra cosilla a tener en cuenta es que José Carlos de Prada usaba en sus TESTs el MS-DOS 3.3 que es más rápido que el MS-DOS 4.0. P.e.: Mi SOLUTION con MS-DOS v4.0 me da 2198 segundos en el TEST2.

En THE SOLUTION cuando escribias algo con el teclado tardaba en salir en pantalla un buen rato. Con CONQUEROR es casi instantaneo, según prioridades. Y los programas como PC TOOL corren más rápido o son más rápidos en dar una respuesta cuando pulsamos una tecla.

Si hay algo que me desilusiono bastante con SOLUTION, fue el hecho de que los primeros programas MS-DOS que pasaron por mis manos, no corrian (e incluso gente que me llamaba por teléfono para decirme que muchos programas no corrian). Eso era más desesperante que la lentitud de los programas en modo emulación. Pero aqui va la buena noticia, ABILITY 2000 y MACADAM (por cierto, una pantalla de máquina de petacos muy bien diseñada, y que sepa, para QL no hay versión. Pero no he conseguido jugar una partida todavía) corren ahora con PC CONQUEROR.

En el caso de ABILITY 2000, en el Solution se quedaba colgado al terminar un meneo de rectangulos que formaban al final la Palabra ABILITY en la pantalla.

Con Conqueror aparece un mensaje de copyright y un mensaje de "pulse una tecla". Luego el programa busca un disco en la unidad B: y lo llena hasta la mitad de no se qué (mi unidad B es una de 5.25" 360K). Luego el programa debe trabajar en modo CGA baja resolución, pues la pantalla de trabajo se hace muy pequeña y las letras me recuerdan el famoso modo 64 columnas del Spectrum. El programa consiste en Base datos, procesador textos, Hoja, gráficos, comunicaciones,.. Lo que se dice un integrado (los gráficos corren bien). La versión observada era en Español y la ayuda funcionaba bien y suficientemente rápido.

Para mi PC CONQUEROR se trata de un juego simulador de ordenadores obsoletos que me sirve para explorar el mundo PC al estilo de Indiana Jones comiendo patatas fritas con refrescos y viendo la TV a la vez (situaciones donde la velocidad no es un factor muy importante).

Hay una pequeña pega con los QLs Españoles frente a uno Inglés (Ernesto me prometió hace casi media dócena de meses su colaboración con su solución, pero se le debe haber olvidado y todavía estoy esperando). Los teclados son diferentes, y para colmo, los códigos ASCII también lo son. Menos mal que el nuevo manual del CONQUEROR es mejor que el de SOLUTION y viene un par de tablas (una corresponde a la configuración por defecto y la otra a la definición de un teclado PC XT). Yo me he entretenido en cambiar las traducciones más importantes para poder usar el CONQUEROR sin estar buscando en el libro QL Programación Avanzada la combinación de teclas correspondiente a la versión MGE de una combinación JM o JS.

El PC CONQUEROR puede correr otros sistemas operativos, además del MS-DOS. P.e.: el XENIX, P-system, CP/M86, ...

Puede usar el disco duro de Miracle Systems de 40 Megas.

En mi configuración, el drive A está configurado para 3.5" 720K y el B para 5.25" 360K.

Para terminar, Digital Precision suele pasarse bastante con la propaganda, pero esta vez su PC CONQUEROR sabiendo que ha sido diseñado para el QL, el Benjamín de los Motorola 680XX, se trata del mejor emulador de IBM PC basado en software escrito hasta el momento.

S. Merino, 29/1/90

LISP

CAPITULO 2.

REPRESENTACION DE CONOCIMIENTOS Y BUSQUEDA DE SOLUCIONES.

Resolución de problemas en inteligencia artificial.

Para que nosotros podamos resolver un problema aplicando las técnicas de IA, éste debe poseer las tres características siguientes:

- Debe entenderse el sistema de representación con el que se plantean los problemas en IA, para poder ser resueltos.
- Deben existir procedimientos de búsqueda de la solución, una vez están representados y almacenados los conocimientos disponibles.
- La solución se ha de poder hallar en un tiempo razonable y debemos poder demostrar que la solución encontrada es correcta.

Para poder aclarar estos conceptos, que son tan importantes en IA, expondré el famoso problema del viajante.

Un viajante debe visitar 4 ciudades: Bilbao, Madrid, Valladolid y Zaragoza. Cada ciudad está unida con las demás por una carretera. Se trata de averiguar cuál es la ruta más corta para visitar todas las ciudades, regresando a la de partida (Fig 1).

```
                    Bilbao
                      *
                    . . .
             278 .    .      . 327
               .      .  366     .
   Valladolid *. . . . . . . . . . * Zaragoza
               .      .          .
             187 .    .401   . 323
                    . . .
```

```
                                 *
    Fig 1.                    Madrid
```

Si las longitudes de las carreteras que se señalan en la fig. 1 no estuviesen expresadas en una unidad conocida (km, m, cm, etc.) sería imposible resolver este problema. Por eso es fundamental definir y entender el sistema de representación de los conocimientos e informaciones con el que se plantea y resuelve el problema. En el caso de la fig. 1 las distancias están expresadas en km.

En segundo lugar, para resolver un problema en IA, debemos utilizar algún procedimiento de búsqueda. Como en el ejemplo del viajante son muy pocas las ciudades a visitar, se pueden probar todas las posibilidades y elegir la más corta. En general, el número posible de rutas para N ciudades sería:

Número de rutas = (N-1)! / 2

En caso de ser N bajo, el número de rutas tambien lo es, pero en cuanto se eleva un poco el valor de N, el número de rutas alcanza cifras impresionantes. En el ejemplo que estamos analizando, es decir N=4, resulta:

Número de rutas = (4-1)! / 2 = 3! / 2 = 3

Es muy importante representar adecuadamente las posibles rutas para facilitar la resolución del problema:

Ruta 1: Bilbao-Valladolid-Madrid-Zaragoza-Bilbao.............1.115 Km.
Ruta 2: Bilbao-Valladolid-Zaragoza-Madrid-Bilbao.............1.368 Km.
Ruta 3: Bilbao-Zaragoza-Valladolid-Madrid-Bilbao.............1.281 Km.

La búsqueda de la solución se complica si las ciudades a visitar son numerosas. Si fueran 10, el número de rutas sería:

Número de rutas = (10-1)! / 2 = 181.440

Con un número de rutas tan grande es lógico que tengamos que aplicar un método de busqueda más sencillo, que sin probar todas las rutas posibles hallase la más corta con el mínimo porcentaje de error, ya que el procedimiento empleado no debe invertir mucho tiempo.

La representación y la búsqueda.

página 1

Para resolver un problema, ya sea en el ordenador o en la mente humana existen tres fases:

1. Definición y representación del problema.
2. Análisis.
3. Elección del método adecuado para resolverlo.

Una buena representación, acompañada de un método de búsqueda eficaz, originan una rápida solución.

A la hora de resolver un problema se presenta la necesidad de elegir entre varias alternativas, las cuales, a su vez, pueden presentar más alternativas, Buscar el camino óptimo entre todas las alternativas en el menor tiempo posible, da origen a los denominados métodos de búsqueda.

En la fig. 2 se muestra un laberinto que debe atravesar un perro situado en la ENTRADA, para alcanzar el hueso de la SALIDA.

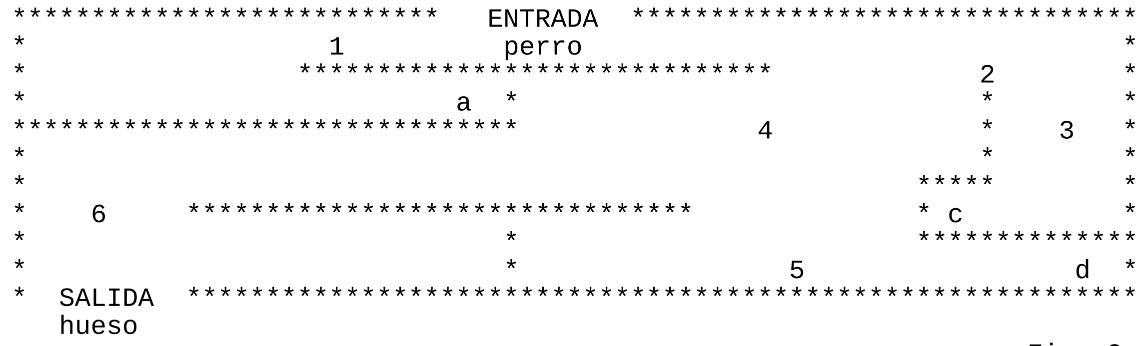


Fig. 2

Para simplificar la búsqueda del hueso en el laberinto, éste se representa en forma de árbol, de manera gráfica, tal como aparece en la fig. 3

```
      perro
     ENTRADA
        *
        .
        .
        *
```

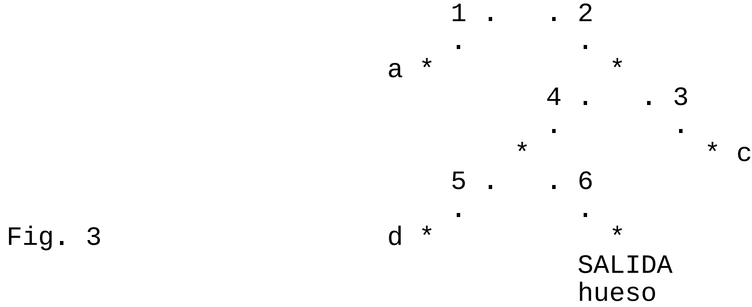


Fig. 3

La fig. 3 aclara enormemente la solución. El perro debe entrar por el camino 2 y luego, siguiendo el 4 y el 6 encontrará el hueso. Cualquier alteración de este recorrido conducirá al perro a un callejón sin salida.

Si el problema que se quiere resolver se define como un espacio de estados, en el que cada uno corresponde a una posible alternativa, se puede resolver comenzando en el estado inicial, aplicándose las diversas técnicas de búsqueda hasta llegar al estado final o solución.

Pero supongamos de nuevo el caso del viajante, pero ahora con más ciudades y con otro planteamiento diferente de su actuación. Las ciudades que pueden componer su ruta se representan en la fig. 4, en forma de esquema, al que se denomina grafo de estados.

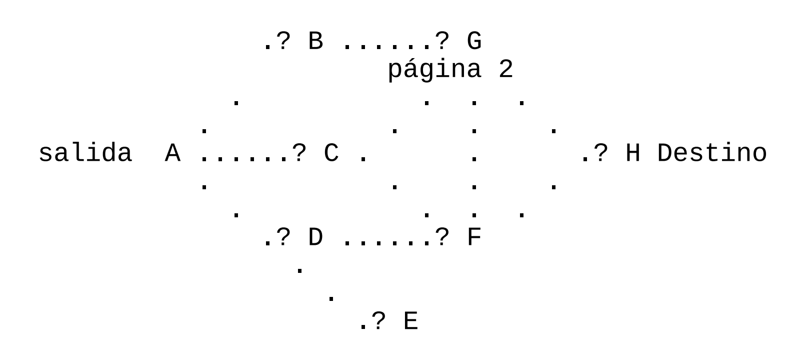


Fig. 4 Ciudades que pueden componer la ruta del viajante y su intercomunicación.

En el ejemplo de la figura 4 se trata de averiguar el orden a seguir desde la salida A hasta alcanzar el destino H.

Un diagrama de espacio de estados se puede representar como un diagrama de árbol.

Arbol es un grafo orientado, en el que cada nodo tiene a lo sumo un predecesor. Hay un nodo que no lo tiene y se llama raiz. Los nodos que no tienen sucesores se llaman hojas y los arcos orientados que unen los nodos entre sí, reciben el nombre de ramas. El árbol que representa el espacio de estados mostrados en la fig. 4 se muestra en la fig. 5.

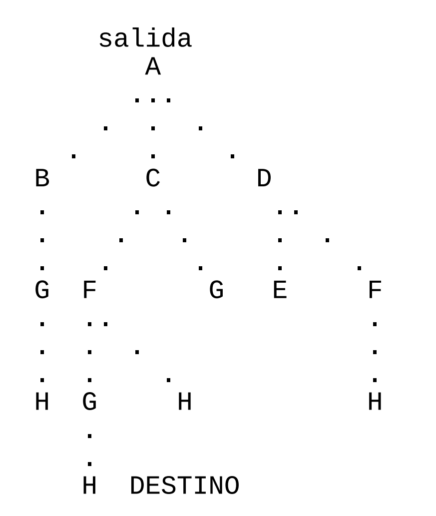


Fig. 5

La representación de las etapas de un problema como un espacio de estados es la base de la mayoría de los métodos de resolución de problemas en los que se

aplican técnicas de IA.

La solución de un problema representado por el diagrama de árbol se alcanza cuando se encuentra el camino que une la raiz y el destino o meta. Para hallar dicho camino se emplean diferentes estrategias de búsqueda, entre las que destacan las siguientes.

1a Búsqueda a lo ancho.
2a Busqueda en profundidad.
3a Busqueda ordenada.

Búsqueda a lo ancho.

Este método consiste en explorar todos los nodos del árbol, nivel por nivel. Hasta que no se hayan explorado todos los nodos del mismo nivel no se pasa al siguiente. El proceso terminará cuando se encuentre el nodo destino.

La búsqueda a lo ancho se puede gestionar mediante una cola. En el final de la cola se van añadiendo los nuevos elementos a analizar y se van retirando los que hay al principio.

En la búsqueda a lo ancho se comienza analizando el nodo raiz, que representa el estado inicial y que, en el caso de la fig. 5 corresponde al nodo A.

Explorando el nodo raiz A, se expande y se colocan sus sucesores o hijos en

página 3

la cola. Los sucesores son los nodos a donde se puede ir desde A. En el caso de la fig. 5, desde A se puede ir a B, C y D. En esta fase de la búsqueda se cierra el nodo A, a lo que gráficamente se simboliza rodeado de asteriscos y se meten en la cola sus hijos, como se refleja en la figura 6.

```
                 *****              entrada de la cola BCD
   nodo cerrado  * A *                   ...........
                 *****                   .    D    .
                  ...                    ...........
                .  .  .                  .    C    .
              .    .    .                ...........
            .      .      .              .    B    .
           B       C       D             ...........
                                       salida de la cola BCD
```

Fig. 6

El siguiente nodo a examinar es el primero de la cola, que en nuestro ejemplo es el B. La cola no es más que una fila donde los nodos que todavía están abiertos, o sea, no explorados, esperan su turno para ser analizados y cerrados. Como el siguiente nodo a explorar, el B, no es el destino buscado, se cierra y se añaden sus hijos a la cola. De esta forma, seguiríamos explorando los nodos que estan al mismo nivel que el B, últimamente despreciado. Debido a la estructura de la cola y su forma de introducir y sacar sus elementos, no se exploran los nodos de un nivel hasta que se hayan analizado los del nivel anterior. Así, iríamos cerrando los nodos hasta pasar al nivel 2, y así en los siguientes niveles. Cuando se detecta en la cola el nodo meta o destino, que en este ejemplo es el H, termina la búsqueda.

Si un nodo aparece varias veces, sólo se expande la primera vez que aparece y en las demás ocurrencias se cierra sin más.

"La búsqueda a lo ancho encuentra el camino más corto desde el nodo inicial hasta la meta.

Búsqueda en profundidad.

Se llama búsqueda en profundidad a la que orienta el análisis de padres e hijos.

En la búsqueda a lo ancho se utilizó la estructura de cola. En cambio en la búsqueda en profundidad, se emplea la estructura de pila.

Para explicar la búsqueda en profundidad nos referiremos al espacio de estados mostrado en la fig. 7.

```
                       .? E
                       .
                       .
                    .? B..........? C
                    .              .
                    .              .
       nodo inicial A..........? D.......? F......? G meta
                    .                .   .       .
                    .        .           .    .
```

```
                    .? H...............? I .
```

Fig. 7

Se comienza explorando el nodo inicial (A) y cuando se cierra, se expande la pila metiendo sushijos, o sea, el B, D y H.

El siguiente nodo a explorar es el B, y después el E. Ahora bien, este nodo es un camino sin salida (hoja), pues de él no sale ninguna rama. En tal caso, se cierra el nodo E y se pasa al siguiente nodo de la pila, que es el C.

El nodo C tampoco tiene salida, luego se cierra y se toma el siguiente de la pila, es decir, el D. Como el D tiene como descendientes el C y el F, éstos se introducen en la pila. Al node C se le cierra por dos razones:

a) Es un callejón sin salida.

b) Ya se había explorado anteriormente.

Del nodo F sólo se puede ir al G, que al introducirse en la pila, se

detectará que es la meta y el proceso se dará por terminado.

Existen varias razones por las que un nodo no se expande:

a) El nodo es un callejón sin salida.

b) El nodo es la meta.

c) Los arcos que salen del nodo en cuestión ya han sido explorados.

d) Por el momento, la búsqueda no ha llegado a ese nodo.

En una búsqueda en profundidad hay casos en los que se asemeja a un pozo sin fin y que la exploración nunca finaliza. Para situaciones como ésta, es corriente poner un límite en la profundidad máxima a alcanzar. Esta restricción presenta un problema, pues puede que se elimine o "pode" la solución idonea (la más corta) o, incluso, que se pode la solución única.

Búsqueda ordenada.

Tanto la búsqueda a lo ancho, como la que se efectúa en profundidad, son búsquedas "a ciegas". El orden en el que se examinan los nodos viene determinado solamente por el tipo de estrategia de búsqueda y no por alguna propiedad o característica de los estados.

Con la búsqueda ordenada se aprovechan las propiedades o información adicional de los estados para averiguar el sentido de la búsqueda.

Se define como costo de un camino en el grafo de estados, el número de operaciones requerido para transformar el primer nodo en el último. Este nodo no es más que el número de ramas o arcos.

Supongamos que se propone estudiar el espacio de estados de la fig. 8

```
                     .? F......? G.
                   .                   .
                 .                          .
Estado inicial A.....? B......? C......? D......? E......? L Estado final
                 .                             .    .    .
                   .                         .       .   .
                     .? H......? I......? J.          .? K
```

FIG. 8

Todos los caminos que van desde el nodo inicial al nodo final tienen un costo. Dicho costo se puede representar en la tabla siguiente.

CAMINO	COSTO
AFGKL	5
AFGEL	4
ABCDEKL	6
ABCDEL	5
AHIJEKL	6
AHIJEL	5

Cada uno de los caminos tiene un costo, pero interesa el de menos costo, que le representaremos con una f.

Supongamos que se analiza el grafo de estados y que se debe decidir cuál de los nodos se ha de explorar. El nodo que se encuentre en el camino solución parece ser la mejor alternativa.

En el ejemplo propuesto se puede deducir el camino solución; sin embargo, cuando se realiza una búsqueda, no se sabe si hay un camino que pase por ese nodo y, menos aún, si es el de menor costo. Por lo tanto, no sabemos el valor exacto de f, pero sí se puede estimar.

Para ello se descompone f en dos partes:

a) Por una, se considera el costo de un camino desde el nodo inicial a un

nodo dado. A este costo se le llama g.

b) Por otra, el costo del camino desde ese nodo hasta el nodo meta, al que se le simboliza por h.

El valor de f será:

$$f = g + h$$

El valor de g es fácil de calcular: el nodo raiz tiene g=0 y los demás nodos tienen una g con un valor una unidad más que el del nivel anterior o, dicho de otro modo, el valor de g es el mismo que el nivel del nodo en el árbol.

página 5

El valor h se puede determinar en el grafo de estados o mediante una regla heurística. Se llama regla heurística aquella que se obtiene por sentido común o por experiencia previa. En el ejemplo propuesto, el valor de h se ha calculado:

NODO	VALOR
F	3
B	4
H	4
G	2
C	3
I	3
D	2
J	2
E	1
K	1

Calculados g y h y por lo tanto f, se puede comenzar la construcción del árbol, para lo cual se empleará una estructura denominada cola de prioridades.

Se empieza por el nodo inicial A que entra en la cola, en la que se introducen los nodos que se van a expandir. En este caso se cierra el nodo A y añadimos sus hijos.

Se calculan los valores f, g y h de los nodos que se acaban de añadir al nodo A y luego se meten en la cola de prioridades ordenados de menor a mayor valor de f. El siguiente nodo a explorar será aquel que tenga menor f. Una vez que se detecta el nodo meta en la cola, finaliza la búsqueda.

Nota: Si el nodo que se está explorando en un momento dado, corresponde al mismo estado que otro nodo, es decir, varios caminos pasan por un mismo nodo, el proceso se complica. Supongamos que tenemos un nodo llamado D ya cerrado y otro nodo D pendiente de explorar que no tiene hijos. ¿Qué hacemos?¿Lo cerramos sin más, como en los demás métodos? No; en este caso nos fijamos en la g de los nodos. Si la g del nodo a explorar es menor que la g del nodo ya cerrado, le quitamos los sucesores al nodo cerrado y se los añadimos al que se está explorando. Volvemos a calcular los valores de f y cambiamos la cola de prioridades.

Conclusiones sobre los métodos de búsqueda.

Hay más métodos de búsqueda por árbol que los que he expuesto, pero el CUQ no tiene 900 paginas y vuestra paciencia tiene un límite. Lo más importante es tener en cuenta que, si podemos suministrar algo de información al programa que está realizando la exploración, para recortar los pasos y para indicar cuál parece ser el siguiente nodo a analizar, se reducirá notablemente el tiempo de ejecución de esta importante fase. La búsqueda a lo ancho y en profundidad puede costar muchos recursos al ordenador.

Un método que optimiza la búsqueda en profundidad es el llamado escalar colinas (Hill climbing).

En este método se lleva a cabo la búsqueda como si estuviesemos empleando la búsqueda en profundidad, pero ordenando los nodos de acuerdo con alguna medida heurística de la distancia que resta. Cuanto mejor sea la medida heurística, mejor resultará la búsqueda.

Esta búsqueda se podrá realizar construyendo una cola con el elemento raiz. A partir de dicho elemento se expanden los nodos de manera que, si el primer elemento de la cola no es el nodo meta, se ordenan los nodos restantes de forma que quede primero aquél al que la regla heurística al que la regla heurística considere el más cercano a la meta. Luego se exploraría ese nodo añadiendo sus hijos a la cola y ordenándoles con el mismo criterio hasta alcanzar la meta.

Los otros métodos existentes se salen fuera del ámbito de este cursillo, dada su complejidad, por cuyo motivo solo los expondré.

En los juegos como el ajedrez, damas, tres en raya, etc., se utiliza el método MINIMAX.

Otro método utilizado en la búsqueda de estados es el ALFABETA.

También son empleados métodos llamados de podas heurísticas, que se basan en

podar o cortar las ramas del árbol, que según la experiencia o enfoque del que ha planeado la búsqueda, se pueden eliminar, ya que no es probable que la
página 6
solución se encuentre en esos caminos. Estos métodos son peligrosos, ya que podemos eliminar caminos válidos.

Es factible que una búsqueda pueda dar lugar a una explosión combinatoria, de la que nunca se puede salir sin emplear un método apropiado o recortar adecuadamente las posibilidades.

En fin, en el próximo número de CUQ intentaré explicaros un poco por encima como puede balbucear un aparato con teclas. Hasta la próxima.

Carlos Iglesias del Río, Madrid, 28-12-89

ASM

HACIENDO COPYS DE GRISES CON 8 Y 24 AGUJAS

Hace ya varias semanas que se me ocurrió escribir los programas assembler que van a adornar el presente articulo. Todo empezó con la compra de la dichosa impresora Epson LQ-550 debido en gran parte a que la BMC estaba pidiendo la jubilación a gritos dandome una calidad de letra horrorosa al negarse a funcionar una aguja correctamente.

Para aquellos que tengan pensamiento de comprarla (aunque hay muchas opciones más interesantes en el mercado), os diré que este bicho acepta todos los modos gráficos y códigos escape de las nuevas y viejas impresoras de 8 y 9 agujas Epson más los nuevos para 24 agujas. Todo ello quiere decir que he probado las configuraciones Epson de la Trump Card con gran éxito.

Como en el gigantesco manual de la LQ-550 solamente he podido encontrar un pequeño resumen de los comandos EPSON, he tenido que buscar la información extra en el viejo manual de la BMC, preguntando a la gente y en un viejo articulo firmado por Orlando Araujo Martin publicado en la desaparecida TODOSPECTRUM-QL MAGAZINE (NUMERO 18 Febrero 1986).

La versión que se expone dentro del articulo corresponde a una versión simplificada y alterada del programa publicado hace años en la mencionada revista en el anterior párrafo.

Como ya os habreis dado cuenta más de uno (en caso contrario mejor tirar la toalla e irnos a llorar a otro sitio), la nueva versión se trata de un programa LBYTES que amplia el Superbasic con el comando IMPRESORA (la versión original era un programa multitarea).

Los resultados son bastante buenos, depende de por donde se mire, en modo 8 obtenemos una copia en pantalla que se chupa todo el folio, pero aunque queda bastante bonito, no se parece en escala mucho al tamaño cuadrado de la pantalla de un QL. En modo 4, nos hace un volcado apaisado (al revés que en modo 8) que ocupa un par de folios (el Cinemascope hemos reinventado). Como ya me decia Nacho por teléfono, hacer un volcado con grises, no es dificil, lo dificil es conseguir acercarse a la escala que usa la pantalla del QL.

Las matrices usadas para representar los colores son :

```
1010100  1000100  1010000  1000010  1000000  0100000  0100000 0000000
0101010  0101010  0100100  0001000  0010100  0001000  0000000 0000000
0010101  0010001  0001010  0100001  0000001  0000010  0000010 0000000

 negro     azul     rojo    magenta  verde    cyan    amarillo  blanco

  11                10                10                        00
  11                01                00                        00
```

Creo que no hace falta explicar más detalles, pues el programa se encuentra

suficientemente comentado. Pero ¡Ojo!, esta versión, como la original publicada en ya se sabe donde, utiliza el modo gráfico Epson 3 y 4, ambos no disponibles en la BMC (solamente usa el modo 1 y 2).

```
           lea.l       define,a1
           move.w      $110,a2     BP.INIT vector
           jmp         (a2)
           rts
*
impresora
           lea         printer,a0  Apunta al nombre de canal
           moveq       #3,d3
           moveq       #-1,d1      canal tomado por DUMP
           moveq       #1,d0
           trap        #2
           tst.l       d0          ¿Estaba ya abierto?
           beq         ok          si es asi continua
           bra         fin
ok
           lea         idp,a1      almacena
           move.l      a0,(a1)     el canal
* Lee el modo de pantalla del QL
           moveq       #-1,d2      lectura del modo
           moveq       #-1,d1      de pantalla
           moveq       #$10,d0
           trap        #1
           tst.b       d1          ¿alta o baja resolución?
           beq         high
           bsr         lo_res      volcado de baja resolución
           bra         done
high       bsr         hi_res      volcado de alta resolución
done       lea         feed12,a1   envia última linea a impresora
           bsr         print       avance de lineas normal
* Cierra el canal de la impresora
           moveq       #2,d0
           trap        #2
fin        rts
*
idp        ds.l        1
* nombre del canal
printer
           dc.w        4
           dc.b        'SER1'
* Datos para avances de linea normales (12 puntos)
feed12     dc.w        4
           dc.b        10,27,'A',12
*
* Volcado a baja resolución
*
* Avance linea de 3 puntos
lo_res     lea         feed3,a1    apunta al dato adecuado
           bsr         print       llama rutina de impresión
* Proceso de datos de pantalla
           movea.l     #$20000,a5  Comienzo del archivo pantalla
           move        #255,d7     256 filas
loop1      move        #63,d6      64 palabras por fila
           bsr         esc         ¿Se ha pulsado ESC?
           lea         mode3,a1    prepara impresora
           bsr         print       7x256 bytes
loop2      move        (a5)+,d4    carga una palabra de pantalla
           moveq       #3,d5       4 pixels por palabra
loop3      moveq       #0,d0       limpia el color
           move.b      d4,d1       * considera palabra impar:
           lsl.b       #1,d1       * cambia los dos bits más
           roxl.b      #1,d0       * significativos.
           lsl.b       #1,d1       * estos determinan los
           roxl.b      #1,d0       * componentes rojo y azul
           lsl         #1,d4       *
           roxl.b      #1,d0
           lsl         #1,d4
```

```
            lsl         #3,d0
            lea         file8,a4    apunta al dato adecuado
            adda        d0,a4
            move        d4,-(a7)    salva la palabra en el stack
            moveq       #6,d4       7 puntos por pixel
loop4       move.b      (a4)+,d1    envia un byte de dato
            bsr         byte        a la impresora
            dbra        d4,loop4    byte siguiente
            move        (a7)+,d4    restablece palabra
            dbra        d5,loop3    pixel siguiente
            dbra        d6,loop2    palabra siguiente
            dbra        d7,loop1    fila sigiente
            rts                     trabajo terminado
* Datos para modo cuadruple densidad de la EPSON
mode3       dc.w        5
            dc.b        10,27,'Z',0,7,0
* Datos para avance de linea de 3 puntos
feed3       dc.w        3
            dc.b        27,'A',3,0
* Modelos para modo de baja resolución (8 colores)
file8       dc.b        %10000000,%1000000      Negro
            dc.b        %10100000,%1000000
            dc.b        %10100000,%1000000
            dc.b        %100000,0
            dc.b        %10000000,0             Verde
            dc.b        %1000000,0
            dc.b        %1000000,0
            dc.b        %100000,0
            dc.b        %10000000,%1000000      Azul
            dc.b        %100000,%1000000
            dc.b        %10000000,%1000000
            dc.b        %100000,0
            dc.b        0,%10000000             Cyan
            dc.b        0,%1000000
            dc.b        0,%100000
            dc.b        0,0
            dc.b        %10000000,%1000000      Rojo
            dc.b        %10000000,%100000
            dc.b        %1000000,%100000
            dc.b        0,0
            dc.b        0,%10000000             Amarillo
            dc.b        0,0
            dc.b        0,%100000
            dc.b        0,0
            dc.b        %10000000,%100000       Magenta
            dc.b        0,%1000000
            dc.b        0,%10000000
            dc.b        %100000,0
            dc.b        0,0                     Blanco
            dc.b        0,0
            dc.b        0,0
            dc.b        0,0
*
* volcado de alta resolución
*
* Avance de linea de 2 puntos
hi_res      lea         feed2,a1    apunta al dato adecuado
            bsr         print       llama rutina de impres.
* Procesa datos de pantalla
            movea.l     #$20000,a5  comienzo archivo pantalla
            move        #63,d7      64 palabras/columna
loop5       move        #7,d6       8 pixels por palabra
loop6       move        #255,d5     256 pixels en una columna
            bsr         esc         ¿ESC pulsada?
            lea         mode4,a1    prepara la impresora
            bsr         print       para 2x256 bytes
loop7       move        d5,d0       indice de columna
            lsl         #7,d0       x1282
            move        0(a5,d0.l),d2     carga una palabra
            moveq       #0,d0       * selecciona
```

```
            moveq       #8,d1       * pixel
            sub         d6,d1       * correcto
            lsl.b       d1,d2       cambia el bit rojo
            roxl.b      #1,d0
            lsl         d1,d2       y el bit verde
            roxl.b      #1,d0
            lsl         #1,d0
            lea         file4,a4    apunta al dato adecuado
            adda        d0,a4
            move.b      (a4)+,d1    envia un byte
            bsr         byte        a la impresora
            move.b      (a4)+,d1    y entonces
            bsr         byte        otro byte
            dbra        d5,loop7    pixel siguiente
            dbra        d6,loop6    palabra siguiente
            addq        #2,a5
            dbra        d7,loop5    columna siguiente
            rts                     trabajo terminado
* Datos para el modo CTR de la EPSON
mode4       dc.w        6
            dc.b        10,27,'*',4,0,2
* Avance de linea de 2 puntos
feed2       dc.w        3
            dc.b        27,'A',2,0
* Modelo de bits para impresora en modo de alta resolución
file4       dc.b        %11000000,%11000000     Negro
            dc.b        %10000000,0             verde
            dc.b        %10000000,%1000000      Rojo
            dc.b        %0,0                    Blanco
* Rutina para enviar cadenas a la impresora
print       move.l      idp,a0       toma la impresora canal ID
            moveq       #-1,d3
            move        (a1)+,d2     N. bytes a enviar
            moveq       #7,d0
            trap        #3
            rts
* Rutina para enviar un byte a la impresora
byte
            move.l      idp,a0       toma la impresora canal ID
            moveq       #-1,d3
            moveq       #5,d0
            trap        #3
            rts
* Comprueba si se ha pulsado ESC
esc         lea         ipc_esc,a3  apunta al comando IPC
            moveq       #$11,d0
            trap        #1
            btst        #3,d1       ¿ESC pulsada?
            bne         abort       si es asi interrumpe
            rts
abort       bsr         beepshort   emite
            move.l      #$ffff,d0   dos pitos  cortos
pause       dbra        d0,pause    como respuesta
            bsr         beepshort
            move.l      (a7)+,a5    cambia ult. dir. retorno
            rts
* rutina pitido corto
beepshort   lea         ipcshort,a3 apunta al comando IPC
            moveq       #$11,d0
            trap        #1
            rts
* comando IPC para pitido corto
ipcshort    dc.b        $a,8        activación del sonido
            dc.l        $aaaa
            dc.b        150,140
            dc.w        0,10
            dc.b        0,0
            dc.b        1,0
* Comando IPC para ESC
ipc_esc
```

```
            dc.b        9,1
            dc.l        0
            dc.b        1,2
define
            dc.w        1            Número de procedimientos
            dc.w        impresora-* relativo a situación de impresora
            dc.b        9            longitud del nombre de procedimento
            dc.b        'impresora' nombre del procedimiento
            dc.w        0            fin de procedimentos
            dc.w        0            número de funciones
            dc.w        0            fin de funciones
            end
```

El ejemplo de volcado en 24 agujas lo he puesto en un fichero aparte ya que se trata de una versión semi alterada de la de 8 agujas. Si en modo 8 agujas se envia un byte que representa 8 agujas en vertical, en modo 24 agujas, se envia 3 bytes seguidos que corresponden a 24 agujas (8 por byte) en vertical de arriba a abajo.

Si alguien desea obtener buenos resultados, aconsejo cambiar la matriz de punteado por otra, según escala que desee obtener. Para ello tendrá que cambiar un par de cosillas: el avance de linea y el número de puntos por pixels.

Si alguno tiene problemas con las pulgadas, en los almanaques de bolsillo normalmente viene una mini regla en pulgadas y cms. Pero yo como buen Andaluz, uso una cinta de 144 pulgadas (Made in UK). Es posible que como siempre, exista alguien con dudas. Acepto preguntas en la sección Cartas Abiertas.

S. Merino, 10/1/1990

LENGUAJE FORTH

DISEÑANDO UN FORTH DE 32 BITS DE DIRECCIONES Y STACK

Desde los tiempos QLaviteros en que os daba la lata con un FORTH de 32 bits, siempre me he temido que terminaria escribiendo mi propio FORTH de 32 bits. Han pasado varios años y poco he conseguido, si no fue el despertar la curiosidad a algunos usuarios de QL que ya eran FORTH parlantes de tiempos atras. Por lo tanto, no he conseguido convencer nunca a nadie, pues quien estaba interesado sabia programar en FORTH.

La verdad es que la serie PRACTICANDO CON EL SUPERFORTH se ha quedado un tanto olvidada, pues nadie ha escrito nada preguntando por ella. Más bien, incluso alguno se ha puesto contento, pero he seguido hablando del FORTH, pero intentando desarrollarlo en el Z88. Y aunque el gran culpable fue Borja Marcos, según el mismo me ha comentado, ha abandonado su intento de implementar el FORTH en el Psion Organiser, porque no tiene ni idea de como se gestiona la memoria en el Psion. Pero en nuestro caso, el Z88 FORTH será una realidad y el mejor lenguaje de programación para el Z88 cuando se encuentre totalmente depurado y todas las utilidades extra añadidas.

Dejemonos de rollo y vamonos a nuestro FORTH de 32 bits. Vamos a preparar el Inner Interpreter (asi lo llaman los Ingleses), el corazón de todo el sistema FORTH (todo lo demás son variables del sistema, diccionario principal y una pequeña rutina que inicia o reinicia el sistema y salta al Outer Interpreter (escrito en FORTH)).

```
*  Inner Interpreter. QDOS FORTH. S.Merino (8/1/1990) @
*  a4 apunta a retorno stack
*  a3 apunta al stack
*  a1 registro de dirección.
*  d1 registro instrucción
*  a2 base del diccionario
*
```

```
*  Pseudocódigo :
*  I   Registro Instrucción
*  LA  Registro dirección Palabra larga
*  CA  Código dirección
*  RS  Retorno Stack
*  SP  Stack
*  PC  Contador Programa
*  @   simboliza el contenido
*
semi       dc.l        semi+4
           move.l      (a4)+,a1    saca RS -> I
next       move.l      (a1)+,d1    @I -> LA    I=I+4
run        move.l      0(a2,d1.l),a5    @LA -> CA
           addq.l      #4,d1            LA=LA+4
           jmp         0(a2,a5.l)       CA -> PC
colon      move.l      a1,-(a4)         pon I -> RS
           move.l      d1,a1            LA -> I
           bra         next             salta a NEXT
*
*
           dc.b        7
           dc.b        "EXECUTE"
           ds.b        24
           dc.l        0
execute    dc.l        execute+4       Código dirección de EXECUTE
           move.l      (a3)+,d1        saca SP -> LA
           bra         run             salta a RUN
```

¡Sorprendidos, nooo..! Todo el Inner Interpreter puede parecer increiblemente complejo o verdaderamente ridiculo. Pero este pequeño código se encarga de mover todo el sistema FORTH.

En nuestro imaginario diseño existen dos tipos de palabras unas llamadas primitivas y las otras secundarias. Las primeras están escritas en código máquina y las segundas en FORTH.

La cabecera de una palabra está formada por un byte que representa el número de letras de la palabra, 31 bytes para almacenar letras, una palabra larga que guarda la dirección de comienzo de la última palabra definida, una palabra larga que guarda la dirección de comienzo del cuerpo de la palabra, y finalmente el cuerpo de nuestra palabra que puede ser tan largo como sea necesario.

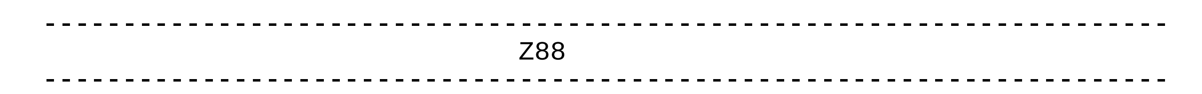
```
       Primitiva                      Secundaria

LA       LA+4    = CA            LA     COLON     =  CA
LA+4       C                     LA+4    LA#1
           O                     LA+8    LA#2
           D                     LA+12   LA#3
           I                               .
           G                               .
           O                               .
                                           .
           M                               .
           A     = CUERPO                  .    =  CUERPO
           Q                               .
           U                               .
           I                               .
           N                               .
           A                               .
LA+N     NEXT    = DIREC. RETORNO   LA+N SEMI   DIREC. RETORNO
```

Es de suponer que todos los que no se han aburrido hasta el momento, y han seguido leyendo, han podido adivinar que existe una pequeña desventaja frente a un FORTH de 16 bits. Las etiquetas o lista de direcciones (el cuerpo de una secundaria, por decirlo más claro que el agua) al ser una dirección de 32 bits ocupa el doble de espacio que la versión de 16 bits, y el código debe correr más lento (especialmente en un QL con un bus de datos de 8 bits).

Aún quedan muchas cosillas por ver, pero es necesario saber el número de interesados, pues según el interés asi de rápido va a ir el desarrollo de este FORTH. Y como no tengo prisa, voy a esperar a recibir en Mayo la documentación del ANS FORTH, pues no tengo decidido aún qué voy a hacer realmente.

S. Merino, 12/1/1990



RANGERDISK DISK DRIVE

Hasta el momento habia 2 sistemas de discos basados en el portátil drive de Tandy y un sistema de almacenamiento en Cassette (Z-Tape). Ahora aparece en escena un nuevo sistema de disco de la mano de Rangers Computers Ltd.

¿Qué ha pasado con el revolucionario sistema de almacenamiento en EPROM? Cuando el Z88 fue lanzado, el precio de los cartuchos EPROM era prohibitivo (En España, 14000 ptas un cartucho Eprom de 128K y 11200 ptas un borrador Eprom. Ambos IVA incluido). Han pasado un par de años y su precio no ha bajado debido en parte a la poca demanda que hay (p.e.: yo mismo todavía no he comprado ningún cartucho EPROM y utilizo el Z88 más horas que el QL). El motivo de la poca demanda es que los programas LINK son muy populares, y existen ya 3 sistemas de disco y uno en Cassette. Como dicen, los usuarios Ingleses, el cómodo sistema almacenamiento en EPROM para viaje ha sido sustituido por un elefante blanco.

El tamaño del RangerDisk es 185x146x67 mm en una caja de metal muy robusto. Pesa 1.6 kg. Es 3.5" IBM compatible 1Mg. Usa 6 pilas del mismo modelo que el Z88 dando una autonomia de 1.5 horas en continua operación.

El software viene en un cartucho ROM de 32K conteniendo el DiskFiler Software. Es muy similar en presentación y uso al FILER.

En velocidad es igual al Disc-88 drive.

La caracteristica principal de este producto es que el formato de disco es IBM compatible (MS-DOS 2.1), pero de momento no soporta directorios en el PC.

La única gran desventaja que tiene este sistema de disco es su precio, 450 libras. Casi el mismo que algunos PC-compatibles de bajo costo.

Ranger Computers, The innovation Centre, Boughton Green Road,
Northampton NN2 7AH.

S. Merino, 29/1/1990

OTROS TEMAS

LAS FUTURAS MAQUINAS PENSANTES... ¿ESPERANTOFONAS DE NACIMIENTO?

La época de los ordenadores "lingüísticos" acaba de empezar. Por todo el mundo, proyectos industriales y universitarios tienden hacia nuevos sistemas mucho más perfeccionados de tratamiento de textos (con correción semiautomática según las leyes de la ortografía, la gramática y el estilo) así como hacia el reconocimiento del lenguaje, la consulta de grandes bases de datos y sistemas expertos por medio del lenguaje natural, la inmediata actualización de diccionarios, etcétera.

Al mismo tiempo, la evolución de potentes sistemas de traducción de nueva generación tiene una importancia estratégica. En medio de la competencia internacional, BSO está fortaleciendo su posición como constructora del DLT (Distribuita Lingvo-Tradukado = Traducción Distribuida de Lenguas), una red "multilengua" entre ordenadores personales.

Esa red futura con un sistema de traducción integrado servirá al nuevo orden internacional de la información, a los paises en vías de desarrollo y aislados lingüísticamente, así como a la comunicación internacional en empresas e instituciones.

Durante los últimos cinco años las inversiones para investigación en DLT no dejaron de crecer. Este año [1987] la suma superará los 6 millones de florines [360 millones de pesetas]. Otros 11 millones [660 millones de pesetas] están previstos para el trienio 1988-1990. La comercialización de DLT comenzará en 1993.

Hasta entonces, habrá que afrontar una dura competencia. De nuestros competidores, algunos pertenecen a las mayores empresas del mundo. Otros tienen un enorme presupuesto gracias a fondos estatales o internacionales, o el apoyo de prestigiosas universidades.

Junto con el sistema boliviano "Atamiri", DLT es el único proyecto que emplea una interlengua completa, y además no está limitado a lenguas indoeuropeas.

El Esperanto es el "as especial" de DLT en la carrera por conseguir la traducción automática.

Ordenadores que puedan comprender la lengua humana, he aquí uno de los más grandes desafíos tecnológicos de hoy. En cierto grado, esa comprensión es necesaria en el proceso de traducción de una lengua a otra, o entre una lengua humana y algún otro sistema formal.

Por ello, la traducción automática y el acercamiento de los ordenadores al idioma natural o a los bancos de información no sólo satisfacerá las crecientes necesidades prácticas, sino que también implica un importante progreso en la teorización y representación del pensamiento humano.

Las redes semánticas son estructuras base generalmente aceptadas en el campo de la Inteligencia Artificial. Las usan, con diversas variaciones, investigadores en EE.UU., Europa y Japón. Cualesquiera que sean los elementos técnicos precisos de estas redes, los conceptos que contienen son sólo presentables mediante palabras de una lengua humana.

En principio cualquier lengua humana podría servir para ello, pero en la práctica el Esperanto es la más adecuada, debido a su alto grado de regularidad y carencia de homónimos.

Así pues, el sistema de traducción DLT se apoya sobre un banco de conocimientos diseñado por completo según el léxico y la semántica del Esperanto. Gracias a un amplio conocimiento mundial, este banco ayuda a reducir la ambigüedad y a enriquecer el producto intermedio del proceso de traducción, es decir la interlengua.

Los primeros éxitos en ese terreno ya han sido alcanzados por BSO en los experimentos de este año [1987]. El éxito final será no obstante el conocimiento mundial general en un disco compacto de DLT, adquirible en todas partes y conectable a los ordenadores caseros y de oficina. Por medio de ese disco se traducirán incluso los subtítulos de la televisión vía satélite.

Esta técnica, perteneciente a una nueva generación, es posible sólo en el marco de un constante aprovisionamiento de talento también perteneciente a una nueva generación... ciertamente también gracias al Esperanto.

Por todo ello deseamos al Esperanto un próspero segundo siglo y una impetuosa decimoprimera década.

BSO/Research
P.K. 8348
NL-3503 RH Utrecht
Holanda

Texto aparecido en el número de julio/agosto de 1987 de la publicación "Esperanto", traducido del Esperanto por:

Marcos Cruz, Madrid, 25/12/1989